



链滴

# Duilib 资源文件详解

作者: [Guozht](#)

原文链接: <https://ld246.com/article/1555442541332>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## Duilib资源文件详解

### 四种分类

```
enum UILIB_RESTYPE
{
    UILIB_FILE=1,    // 来自磁盘文件
    UILIB_ZIP,       // 来自磁盘zip压缩包
    UILIB_RESOURCE,  // 来自资源
    UILIB_ZIPRESOURCE, // 来自资源的zip压缩包
};
```

### UILIB\_FILE形式加载资源

这种方式会加载存放在硬盘中的资源。

具体使用方式

```
CPaintManagerUI::SetResourceType(UILIB_RESOURCE);//设置资源类型
CDuiString strResourcePath = CPaintManagerUI::GetInstancePath();
strResourcePath += _T("skin\\");
CPaintManagerUI::SetResourcePath(strResourcePath.GetData());
//设置资源路径
```

### UILIB\_ZIP形式加载资源

这种方式会加载存放在硬盘中的压缩包资源。

具体使用方式

```

CPaintManagerUI::SetResourceType(UILIB_ZIP);//设置资源类型
CDuiString strResourcePath = CPaintManagerUI::GetInstancePath();
strResourcePath += _T("skin\\");
CPaintManagerUI::SetResourcePath(strResourcePath.GetData());
//设置资源路径
CPaintManagerUI::SetResourceZip(_T("skin.zip"), true);//设置压缩包资源, 可以加上第三个密码数 (默认为NULL)

```

## UILIB\_RESOURCE形式加载资源

这种方式会加载可执行程序中的资源文件。

具体使用方式

```

CPaintManagerUI::SetResourceType(UILIB_RESOURCE);//设置资源类型

```

## UILIB\_ZIPRESOURCE形式加载资源

这种方式会加载可执行程序中的zip压缩文件。

具体使用方式

```

CPaintManagerUI::SetResourceType(UILIB_ZIPRESOURCE);//设置资源类型
HRSRC hResource = ::FindResource(CPaintManagerUI::GetResourceDll(), _T("IDR_ZIPRES"), _T("ZIPRES"));
if (hResource != NULL) {
    DWORD dwSize = 0;
    HGLOBAL hGlobal = ::LoadResource(CPaintManagerUI::GetResourceDll(), hResource);
    if (hGlobal != NULL) {
        dwSize = ::SizeofResource(CPaintManagerUI::GetResourceDll(), hResource);
        if (dwSize > 0) {
            CPaintManagerUI::SetResourceZip((LPBYTE)::LockResource(hGlobal), dwSize);
        }
    }
    ::FreeResource(hResource);
}

```

## 运作原理

两种ZIP资源会调用 CPaintManagerUI的SetResourceZip函数将资源加载保存

```

void CPaintManagerUI::SetResourceZip(LPCTSTR pStrPath, bool bCachedResourceZip, LPCTSTR password)
{
    if ( m_pStrResourceZip == pStrPath && m_bCachedResourceZip == bCachedResourceZip )
return;
    if( m_bCachedResourceZip && m_hResourceZip != NULL ) {
        CloseZip((HZIP)m_hResourceZip);
        m_hResourceZip = NULL;
    }
    m_pStrResourceZip = pStrPath;
    m_bCachedResourceZip = bCachedResourceZip;
    m_pStrResourceZipPwd = password;
    if( m_bCachedResourceZip ) {

```

```

    CDuiString sFile = CPaintManagerUI::GetResourcePath();
    sFile += CPaintManagerUI::GetResourceZip();
#ifdef UNICODE
    char* pwd = w2a((wchar_t*)password);
    m_hResourceZip = (HANDLE)OpenZip(sFile.GetData(), pwd);
    if(pwd) {
        delete[] pwd;
        pwd = NULL;
    }
#else
    m_hResourceZip = (HANDLE)OpenZip(sFile.GetData(), password);
#endif
}
}

```

之后在使用字体或图片等资源时会进行类似下面加载图片的判断，

首先判断type是否为空（type为XML文件之内控件的restype属性），

如果type不为空则认为是可执行程序中的资源文件，之后将图片从资源中取出。

如果type不为空，则继续判断之前是否保存过ZIP文件名（UILIB\_ZIPRESOURCE调用CPaintManagerUI::SetResourceZip(...)时会默认将ZIP文件名保存为mемbuffer），

如果保存过ZIP文件，则加载zip中图片资源，

如果没有保存过，则在磁盘中加载文件。

TImageInfo\* CRenderEngine::LoadImage(STRINGorID bitmap, LPCTSTR type, DWORD mask, HINSTANCE instance)

```

{
    LPBYTE pData = NULL;
    DWORD dwSize = 0;
    do
    {
        if( type == NULL ) {
            CDuiString sFile = CPaintManagerUI::GetResourcePath();
            if( CPaintManagerUI::GetResourceZip().IsEmpty() ) {
                sFile += bitmap.m_lpstr;
                HANDLE hFile = ::CreateFile(sFile.GetData(), GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, \
                    FILE_ATTRIBUTE_NORMAL, NULL);
                if( hFile == INVALID_HANDLE_VALUE ) break;
                dwSize = ::GetFileSize(hFile, NULL);
                if( dwSize == 0 ) break;

                DWORD dwRead = 0;
                pData = new BYTE[ dwSize ];
                ::ReadFile( hFile, pData, dwSize, &dwRead, NULL );
                ::CloseHandle( hFile );

                if( dwRead != dwSize ) {
                    delete[] pData;
                    pData = NULL;
                    break;
                }
            }
        }
    }
}

```

```

else {
    sFile += CPaintManagerUI::GetResourceZip();
    CDuiString sFilePwd = CPaintManagerUI::GetResourceZipPwd();
    HZIP hz = NULL;
    if( CPaintManagerUI::IsCachedResourceZip() ) hz = (HZIP)CPaintManagerUI::GetRes
ourceZipHandle();
    else
    {
#ifdef UNICODE
        char* pwd = w2a((wchar_t*)sFilePwd.GetData());
        hz = OpenZip(sFile.GetData(), pwd);
        if(pwd) delete[] pwd;
#else
        hz = OpenZip(sFile.GetData(), sFilePwd.GetData());
#endif
    }
    if( hz == NULL ) break;
    ZIPENTRY ze;
    int i = 0;
    CDuiString key = bitmap.m_lpstr;
    key.Replace(_T("\\"), _T("/"));
    if( FindZipItem(hz, key, true, &i, &ze) != 0 ) break;
    dwSize = ze.unc_size;
    if( dwSize == 0 ) break;
    pData = new BYTE[ dwSize ];
    int res = UnzipItem(hz, i, pData, dwSize);
    if( res != 0x00000000 && res != 0x00000600 ) {
        delete[] pData;
        pData = NULL;
        if( !CPaintManagerUI::IsCachedResourceZip() ) CloseZip(hz);
        break;
    }
    if( !CPaintManagerUI::IsCachedResourceZip() ) CloseZip(hz);
}
}
else {
    HINSTANCE dllinstance = NULL;
    if (instance) {
        dllinstance = instance;
    }
    else {
        dllinstance = CPaintManagerUI::GetResourceDll();
    }
    HRSRC hResource = ::FindResource(dllinstance, bitmap.m_lpstr, type);
    if( hResource == NULL ) break;
    HGLOBAL hGlobal = ::LoadResource(dllinstance, hResource);
    if( hGlobal == NULL ) {
        FreeResource(hResource);
        break;
    }

    dwSize = ::SizeofResource(dllinstance, hResource);
    if( dwSize == 0 ) break;
    pData = new BYTE[ dwSize ];

```

```
        ::CopyMemory(pData, (LPBYTE)::LockResource(hGlobal), dwSize);
        ::FreeResource(hResource);
    }
} while (0);
.....
}
```